PATENT COOPERATION TREATY

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

From the INTERNATIONAL BUREAU

To:

Assistant Commissioner for Patents
United States Patent and Trademark Office
Box PCT
Washington, D.C.20231
ETATS-UNIS D'AMERIQUE

in its capacity as elected Office

Date of mailing (day/month/year) 14 August 2000 (14.08.00)	
International application No. PCT/IB99/01989	**Applicant's or agent's file reference** F14839
International filing date (day/month/year) 14 December 1999 (14.12.99)	**Priority date** (day/month/year) 18 December 1998 (18.12.98)

Applicant

GORDON, Graham, Paul

1. The designated Office is hereby notified of its election made:

   [X] in the demand filed with the International Preliminary Examining Authority on:

   03 July 2000 (03.07.00)

   [ ] in a notice effecting later election filed with the International Bureau on:

2. The election [X] was

   [ ] was not

   made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland	**Authorized officer** Zakaria EL KHODARY
Facsimile No.: (41-22) 740.14.35	Telephone No.: (41-22) 338.83.38

2640 Pitcairn Road
Monroeville, Pennsylvania 15146

412 · 380 · 0725
Facsimile 412 · 380 · 0748

James O. Ray, Jr. *

Forest C. Sexton **
Edmond S. Miksch *
James R. Stevenson ***
Elroy Strickland †
Amos Bartoli *
Michele K. Yoder *
Robert A. Shack ††

* *reg US Patent Agent*
** *admitted to PA Bar, U.S. Dist.CT., W.PA*
*** *admitted to PA & MI Bar*
† *admitted to D.C., CAFC, Sup. Ct.*
†† *admitted to PA & MA Bar*

Of Counsel

Alfred D. Lobo
admitted to practice OH & CAFC Bar

Foreign Patent Manager

Todd A. Ray

Docket ADM-01363

Date: 14 June 2001

Commissioner of Patents
Washington, D.C. 20231

Attn: Box PCT

RE: National Stage Entry in the United States for PCT International
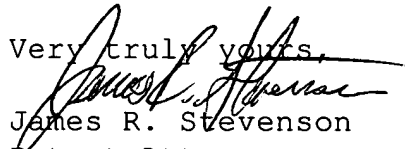Application PCT/IB99/01989.

Dear Sir/Madam:

Pursuant to the Patent Cooperation Treaty (PCT), we request that the
aforementioned PCT International Application be entered into the
national stage in the United States. To that end, we submit the
following items to the U.S. Patent and Trademark Office (USPTO):

(1)   *PCT Transmittal Letter* (Form PTO-1390) (2 sheets);
(2)   *PCT International Application PCT/IB99/01989* (i.e. WO 00/38051);
(3)   *International Search Report* (PCT/ISA/210) (2 sheets);
(4)   *International Preliminary Examination Report* (PCT/IPEA/409) (6 sheets)
      with substitute pages 2 and 2A and claims 1-6 attached (5 sheets);
(5)   *Statement Claiming Small Entity Status* (1 sheet);
(6)   A *Declaration of Inventor(s) And Power of Attorney* form;
(7)   A *Preliminary Amendment;* and
(8)   A check for $430.00 to cover the *filing fee* for a small entity.

If you have any questions about the matters discussed in this
correspondence, please contact our office accordingly.

Very truly yours,

James R. Stevenson
Patent Attorney

Enclosures: as stated

JRS:sf

THIS PAGE BLANK (USPTO)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

| (51) International Patent Classification 7 : G06F 9/44 | A1 | (11) International Publication Number: WO 00/38051 |
| | | (43) International Publication Date: 29 June 2000 (29.06.00) |

(21) International Application Number: PCT/IB99/01989

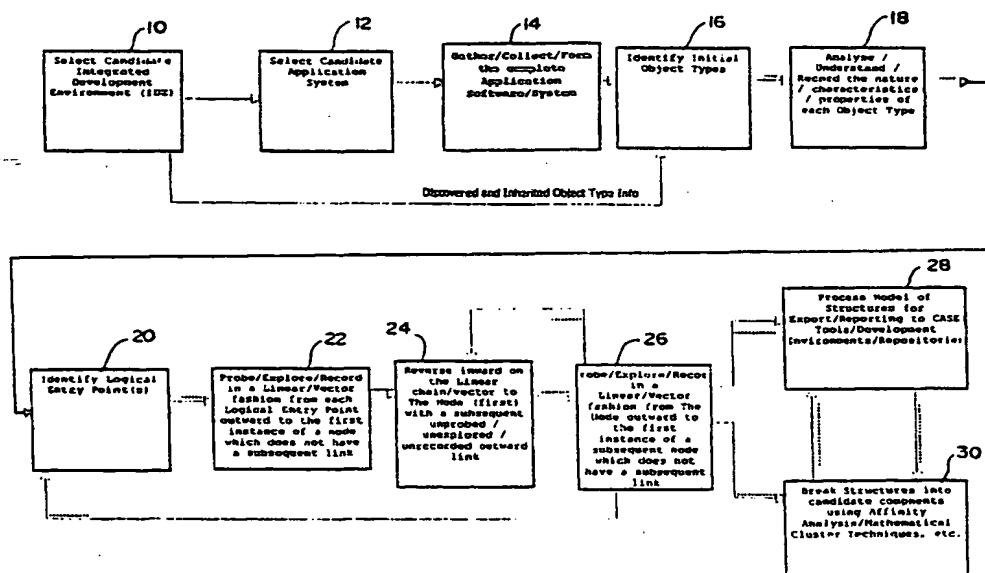(22) International Filing Date: 14 December 1999 (14.12.99)

(30) Priority Data:
98/11657   18 December 1998 (18.12.98)   ZA

(71)(72) Applicant and Inventor: GORDON, Graham, Paul [ZA/ZA]; Overport, 4067, 519 Moore Road, P.O. Box 70005, Glenwood, 4001 Durban (ZA).

(74) Agent: ROTTEVEEL, Martin; Adams & Adams, Adams & Adams Place, 16 Cranbrook Crescent, La Lucia Ridge, Office Park, Umhlanga, 4320 Umhlanga (ZA).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: A METHOD OF PERFORMING A SYSTEM REVERSE ENGINEERING PROCESS

(57) Abstract

The invention relates to a method of performing a system reverse engineering process. The method provides for an examination of the application system that requires reverse engineering by examining the entire network structure forming the system, by tracking chains of nodes and links in accordance with a predetermined tracking method, that will ensure a complete examination. The information gathered from the examination will permit formatting of the information into a form in which it represents the application system in a usable form. The method of the invention provides also for the creation of a software program or an entire software system that can be employed for carrying out the reverse engineering process.

THIS PAGE BLANK (USPTO)

A METHOD OF PERFORMING A SYSTEM REVERSE ENGINEERING PROCESS

THIS INVENTION relates to a method of performing a system reverse engineering process.

It is known that as a result of software system accretion, which occurs when systems are linked together, when systems are built on or hacked into in a relatively
5   unstructured mode to effect quick fixes or changes, when systems are modified to accommodate other systems and/or the like, a conglomerate system can result which, for various reasons, can no longer be easily managed and which is not understood in all respects. In extreme cases this can result in a system becoming obsolete, inutile, or too complicated to continue to operate and work with, essentially requiring
10 system replacement.

The latter option often is not economically or technically feasible and in order to at least alleviate the problem identified and for various other reasons, system reverse engineering processes have been developed whereby conglomerated systems can be re-engineered into a workable format. System reverse engineering processes involve
15 essentially the examination of the existing system, the documentation of the system, modelling of the system, analyzing of the system and understanding of the system, whereafter it is possible to re-engineer the system into a workable and useful format.

A system that requires to be reverse engineered as herein envisaged, hereinafter referred to as the application system, comprises a network structure of nodes and
20 links, the nodes and links forming chains that either terminate in a final node or that form a closed loop that extends from a node and returns to the same node. Network

structures are further complicated insofar as two or more links can extend from a node and by keeping in mind that a network structure could include millions of nodes and links, it will be appreciated that very intricate structures can result. The individual nodes and links referred to essentially are object instances and messages

5   disposed between the object instances, representing activity and data elements which are associated with the operation of the system, for carrying out its required purpose or purposes.

In order to apply a reverse engineering process to an application system, it is required to obtain a full understanding of the system, i.e. an understanding of the operation

10   of the network structure forming the system, in order to permit the system to be reverse engineered into a format which permits a model of the system to be created with the aid of a suitable CASE tool (Computer Aided Software/Systems Engineering) or any other categories of suitable visualisation tools. The examination of an application system in order to acquire an understanding of the system conventionally

15   involves an overall consideration of the system and then progressively delving into the system from a number of predetermined starting points, delving deeper and deeper into the system until the required understanding is acquired. This generally requires the cooperation of a team of suitably qualified systems engineers who will cooperate with one another and add their knowledge together until the required level

20   of understanding of the system is acquired, which then permits reverse engineering. This examination system is well known and produced desired results in relation to relatively simpler and less dynamically changing application systems where systems engineers could acquire a sufficient overall picture of the system to permit reverse engineering thereof, but in relation to more complex and dynamic systems this

25   method of examination became too complex and time consuming and, as such, impractical.

The more complex application systems requiring reverse engineering therefore cannot be economically salvaged, even with the aid of software programs assisting with the examination process as above envisaged. As such, it is an object of this invention

30   to provide an improved method of performing a system reverse engineering process

which will permit more complex application systems to be reverse engineered to a workable format and to make reverse engineering of less valuable systems commercially viable.

According to the invention there is provided a method of performing a system reverse

5   engineering process, which includes the steps of

identifying the application system that requires reverse engineering and gathering the entire system and identifying the development environment associated with the system;

identifying initial object types that can serve as starting points from where an

10   examination of the system can be initiated and analyzing the nature, characteristics and properties of each object type;

identifying entry points, in the form of object instances of certain object types, for entering the system to carry out the examination of the system;

examining from selected entry points the network structure forming the

15   application system by tracking chains of nodes and links, each chain being tracked until the instance of a node that does not have a link or the return of the chain to a previously examined node, then reverse tracking the chain to a node from which another chain extends and selectively tracking said other chain and continuing the process until all the chains within the network structure have been tracked, the

20   tracking of the chains including an examination of each node and link and a recordal of information so gathered; and

from the information gathered by the network examination, formatting the information gathered into a form in which it represents the application system in a usable form.

In identifying the application system that requires reverse engineering and gathering the entire system, it must be ensured that the important components of the system are taken account of in order to ensure the effectiveness of the reverse engineering process. It must be appreciated also that the application system that requires reverse engineering may comprise two or more separate systems having common elements, or a conglomeration of two or more systems. Any reference herein to an application system must be interpreted as such.

In identifying the development environment, which may be an integrated development environment, the method of the invention may include identifying aspects of the development environment selected from a group including program language and syntax used, the mechanisms of storage of data, the interface of the above, component libraries, code management systems, and the like.

The object types identified typically may fall into multiple categories or groups, including process or activity control elements, data management elements and interface elements. The object types within these groups generally are manifest in nodes and links which are in the form of object instances and messages and which form the overall network structure representing the application system. It is believed in this regard that the vast majority, if not all, development environments ranging from old legacy development environments to modern object oriented integrated development environments and any application systems built therein are essentially networks of nodes and links or objects and messages which can be modelled or described by notations such as the Unified Modelling Language (UML) inter alia but not limited thereto.

The examination of the network structure comprises an examination of each node and each link in the structure to the extent that the nature, characteristics and properties of each node and each link can be associated with object types through analysis, matching, mapping and understanding thereof, and gathering and recording all the information of each node and each link, which may include inheriting information from known or predefined object type information about the development environment.

The entry points identified therefore may comprise the nodes from which a complete examination of the entire network structure of the application system can be initiated. The examination of the network structure also involves the complete tracking of each chain to its termination or return to an earlier node in the chain, before a further chain

5   is selected and tracked. The tracking of chains within the examination of the network structure may include a comparison and classification of nodes and links as object instances, to establish whether they conform with known or unknown object types.

The information gathered from the examination of the network structure will enable a complete understanding of the network structure and particularly also its object

10   types and their object instances, which in turn will permit formatting of this information into a logical format in which the application system is again rendered usable.

The method of the invention particularly provides for formatting of the information gathered into a format in which the information can be exported/reported to CASE

15   tools, development environments and/or repositories, enabling the creation of a model of the application system. As such, formatting may include breaking structures into candidate components by using affinity analysis, mathematical clustering techniques, and the like.

The method of the invention provides still further for the employment of software

20   and/or hardware for assisting with the identification of object instances of object types and the analysis of the nature, characteristics, attributes, operations and properties of each object instance and object type identified, the identification of entry points for entering the system to carry out the examination of the network structure forming the system, the actual examination of the network structure and

25   the formatting of the information gathered by the examination of the network structure into a usable form.

As such, the method of the invention includes creating a software program or an entire software system that can be employed for the above purpose in respect of the

particular application system being reverse engineered. A software program, as envisaged, must be interpreted to include any substitute for such a program, which may be a hardware component, or the like.

Furthermore, in relation to the employment of the software program as a result of
5   which object types are identified which were not originally accounted for, the method of the invention may include modifying the software program in order to take into account the object types so identified. This may include also modifying the base of object type information already gained which may be described in the UML but not limited thereto.

10   It will be understood in the above regard that although it is the employment of the software program that renders the method practically feasible, particularly in relation to more complex application systems, it is the method steps as defined and which must be followed, which renders the use of a software program for the purpose practically feasible.

15   The method of performing a system reverse engineering process, in accordance with the invention, can be used in respect of a wide range of application systems that are associated with the problems hereinabove identified, the method of the invention essentially enabling these application systems to be salvaged by re-formatting of the systems through the reverse engineering thereof into a form in which the systems
20   again be made practically usable.

The method of performing a system reverse engineering process, in accordance with the invention, is described hereinafter with reference to the accompanying diagrams. In the diagrams:

      Figure 1 illustrates diagrammatically in block diagram form a flow chart
25   illustrating the method of performing a system reverse engineering process, in accordance with the invention;

Figure 2 illustrates diagrammatically in block diagram form a flow chart setting out a non-application system process for illustrating the method of performing a system reverse engineering process, in accordance with the invention; and

5   Figure 3 (13 pages) illustrates diagrammatically in block form a particular example of a method of performing a system reverse engineering process, in accordance with the invention, referring by way of explanation to the block diagram illustrated in Figure 2.

Referring initially to Figure 1 of the drawings, a method of performing a system reverse engineering process, in accordance with the invention, is illustrated as a flow

10   chart in block diagram form. Blocks 10, 12 and 14 represent the initiation of the method which includes the selection/identification of the application system that requires reverse engineering, the gathering of all the components of the application system which are required for the operation of the entire system and the identification/selection of the development environment associated with the system.

15   This development environment may be an integrated development environment which includes the program language and syntax used, the mechanisms of storage of data and the interface of the above. Clearly, the development environment also may include other aspects which are associated directly with the application system involved such as component libraries and code management systems.

20   Blocks 16 and 18 represent the method steps of identifying initial object types incorporated within the system and that can serve as starting points from where an examination of the system must be initiated and analyzing and understanding the nature, characteristics and properties of each object type to enhance still further the overall understanding of the system and the associated development environment.

25   It must be appreciated in this regard that the object types identified essentially will fall into multiple categories or groups, i.e. a first group may include process, activity or control elements, a second group may include data management elements and a third group may include interface elements. Examples of object types falling within the above three groups are set out below:

Examples of Process, Activity or Control Elements as Object Types

        Program
        Procedure
        Sub-Procedure
5   Library Procedure
        Class
        Call
        Invocation
        Message
10   Command/Verb
        Statement
        Algorithm Flow/Control
        Rules/Conditions
        Method
15   Operation
        Service

Examples of Data Management Elements

        Table
        Field
20   File
        Entity
        Attribute
        Relationship
        Relation
25   Array
        Variable
        Parameter
        Pointer
        Message

30   Examples of Interface Elements ("things" the user of the application system sees or interacts with or interface elements to other systems.)

        Dialog
        Report
        Screen (Read-only, Read-write)
35   Menu
        Window
        List
        Button
        Text Box
40   Check Box
        Radio Button
        Tree

It will be understood that additional object types falling within the above groups may be implemented in object instances within an application system and, as set out hereafter, these object types, once identified through the method of reverse engineering as hereinafter described, can then be categorized on an ad-hoc basis. It

5   must also be understood at this stage that the object types referred to above generally are acting as nodes and links which form the overall network structure representing the application system in respect of which reverse engineering is required, the nodes and links forming chains which themselves define the network structure.

10   The next step in the method of performing a system reverse engineering process is represented by block 20 and involves the identification of logical entry points through which the system can be entered for examination purposes.

Once these entry points have been identified, the next step within the method of the invention involves the examination of the network structure forming the system,

15   which includes selecting entry points from the logical entry points already identified and tracking the chains of nodes and links extending from these entry points, each chain being tracked until the instance of a node that does not have a link or the return of the chain to a previously examined node. This is then followed by a reverse tracking of the chain to a node from which another chain extends, selectively tracking

20   the said other chain and continuing the process in the manner defined until all the chains within the network structure have been tracked, the tracking of the chains including also an examination of each node and link and a recordal of information gathered from this examination of each node and link. It must be understood that when reverse tracking of a chain is referred to, a reverse path along a chain will be

25   followed until a node is identified from which another chain extends that will then be tracked, unless it has already been tracked. If a node or link of unknown type is hit which requires first to be defined, this will be recorded or logged for subsequent attention and reverse tracking will continue until a node is identified from which a chain extends which will permit tracking. The overall objective remains that

substantially all the chains within a network structure and particularly all the nodes and links within the structure should be examined and information in respect thereof must be gathered, although nodes and links identified as unimportant or irrelevant

5   could be ignored. In Figure 1 the blocks 22, 24 and 26 represent this examination process, at the completion of which a complete understanding of the original application system should be possible insofar as all the nodes and links forming chains within the network structure representing the system will have been examined and the properties, functions, attributes, operations and characteristics of the nodes and links will be known.

10   The final step in the reverse engineering method of the invention hence involves formatting of the information gathered by the examination referred to above, particularly using the information gathered for formatting the application system in a network structure form which is effectively usable, i.e. in a form in which the application system is understandable and the system can again be used for fulfilling

15   its required purpose in a normal manner, while also permitting the application system to be worked with and modified as may be required from time to time. This latter step in the method of forming a system reverse engineering process is represented by the blocks 28 and 30 from which it will be appreciated that the newly formatted application system will be in a form in which predetermined CASE tools and modelling

20   languages can be utilised for creating a model of the application system, if required.

In order to facilitate the application of the method of performing a system reverse engineering process, in accordance with the invention, the method steps associated with the blocks 16 to 30 can be carried out with the aid of a suitable software program or entire software system that has been created for the purpose and

25   particularly for use in conjunction with the development environment of the application system to be reverse engineered. The creation of this software program or entire software system accordingly also may form a part of the method of the invention, it being envisaged in this regard that it may be required to modify the software program from time to time, or to expand the base of "known" object type

30   information, as a result of unknown object types being identified during the

examination process, permitting a systems engineer to categorise the object type and then provide the necessary information within the software program and/or "known" base in order to deal with this object type in carrying out the method of the invention as described. In this regard it will be appreciated that the method of the invention

5   could be "manually" carried out in relation to relatively simple application systems, but in relation to practical application systems which do in fact require reverse engineering, the assistance of a suitable software program will be essentially required.

Referring to Figure 2 of the drawings, in order to explain the method of performing a system reverse engineering process, the process can be equated to the examination

10   of a building having a plurality of rooms that are interlinked with one another in an essentially random fashion via doors between them and where the layout of the building is not known and therefore requires examination. In this diagram, block 40 can be associated with the identification of the application system to be reverse engineered, while block 42 provides for the identification of entrance doors which can

15   lead into the system/building for examination purposes. Different selected entrance doors will then be entered by different members of the examination team involved.

The examination of the rooms within the building as represented by blocks 44, 46, 48, 50, 52, 54, 56, 58, 60, 62 and 64, will effectively represent the examination process associated with the method of the invention insofar as rooms will be

20   examined one after the other, until a room is reached which does not have a further door therein, following which the chain followed will be reversed until a room is reached from where another door extends and from where the examination process can continue. The process as described by the blocks 44 to 64 clearly will result in each room in the building being examined, which will in fact equate to the

25   examination of all the nodes and links within a network structure of an application system.

All the information gathered from the individual rooms will then be set out in a logical format, which operation is represented by the block 66, this logical format enabling one to obtain a clear picture of the layout of the building involved. This layout of the

building, which will now be clear, will equate to the new format of the application system that has been created, which will comprise an understandable format which will give a clear understanding of the application system and, particularly, the network structure forming the system.

5　This new format, insofar as it applies to the building examined, will then permit additions to the building to be effected or a model of the building to be built and this clearly equates to the use of the new format in relation to the method of the invention which permits the creation of a model of the application system involved with the aid of suitable CASE tools, or merely the normal application of the application system

10　which may require system modifications, additions and the like, which could again be logically carried out as a result of the complete understanding of the system which is acquired through the reverse engineering process as described.

Referring to Figure 3 of the drawings, a typical example of a comprehensive automated reverse engineering method which includes the employment of the method

15　of the invention is described in a block diagram form and in association with the simulation of the method of the invention as illustrated in Figure 2 of the drawings. The individual steps as illustrated in Figure 3 of the drawings therefore are cross-referenced as step numbers with reference to Figure 2 of the drawings, the step numbers being associated with the numbers 1 to 10 included within the blocks

20　forming the diagram illustrated in Figure 2 of the drawings. As the method of the invention in its application with reference to Figure 3 will be clear to those skilled in the art, the method as illustrated in Figure 3 of the drawings is not described in detail hereafter.

It must be appreciated that the method of the invention can be applied in association

25　with many different application systems that require reverse engineering, essentially enabling salvaging of application systems which may otherwise have become obsolete, insofar as the application systems will be formatted into forms in which the systems are again rendered usable through the effective understanding of the systems.

CLAIMS

1.   A method of performing a system reverse engineering process, which includes the steps of :

    identifying the application system that requires reverse engineering and
5   gathering the entire system and identifying the development environment associated with the system;

    identifying initial object types that can serve as starting points from where an examination of the system can be initiated and analysing the nature, characteristics and properties of each object type;

10   identifying entry points, in the form of object instances of certain object types, for entering the system to carry out the examination of the system;

    examining from selected entry points the network structure forming the application system by tracking chains of nodes and links, each chain being tracked until the instance of a node that does not have a link or the return of
15   the chain to a previously examined node, then reverse tracking the chain to a node from which another chain extends and selectively tracking said other chain and continuing the process until all the chains within the network structure have been tracked, the tracking of the chains including an examination of each node and link and a recordal of information so gathered;
20   and

    from the information gathered by the network examination, formatting the information gathered into a form in which it represents the application system in a usable form.

2.   A method as claimed in Claim 1, in which the development environment
25   identified is an integrated development environment.

3.   A method as claimed in Claim 1 or Claim 2, which includes in identifying the development environment, identifying aspects of the development environment selected from a group including program language and syntax used, the mechanisms of storage of data, the interface of the above, component libraries and code management systems.

4.   A method as claimed in any one of Claims 1 to 3, in which the object types identified fall into multiple groups, respective groups including process or activity control elements, data management elements and interface elements.

5.   A method as claimed in any one of the preceding claims, which includes examining the network structure by an examination of each node and each link in the structure to the extent that the nature, characteristics and properties of each node and each link can be associated with object types through analysis and understanding thereof and gathering and recording all the information of each node and each link.

6.   A method as claimed in any one of the preceding claims, which includes identifying entry points that comprise the nodes from which a complete examination of the entire network structure of the application system can be initiated.

7.   A method as claimed in any one of the preceding claims, which includes examining the network structure by the complete tracking of each chain to its termination or return to an earlier node in the chain, before a further chain is selected and tracked.

8.   A method as claimed in any one of the preceding claims, in which the tracking of chains within the examination of the network structure includes a comparison and classification of nodes and links as object instances to establish whether they confirm with known or unknown object types.

9.   A method as claimed in any one of the preceding claims, which includes
      formatting the information gathered into a format in which the information can
      be exported/reported to at least one of a predetermined computer aided
      software/systems engineering tool, a development environment and a
5   repository, which will enable the creation of a model of the application system.

10.   A method as claimed in Claim 9, in which formatting includes breaking
      structures into candidate components by using affinity analysis and
      mathematical clustering techniques.

11.   A method as claimed in any one of the preceding claims, which includes
10   employing any one of software and hardware for assisting with the
      identification of object instances of object types and the analysis of the nature,
      characteristics, attributes, operations and properties of each object instance
      and object type identified, the identification of entry points for entering the
      system to carry out the examination of the network structure forming the
15   system, the actual examination of the network structure and the formatting of
      information gathered by the examination of the network structure into a usable
      form.

12.   A method as claimed in Claim 11, which includes creating a software program
      or entire software system which can be employed in respect of the particular
20   application system and associated development environment being reverse
      engineered.

13.   A method as claimed in Claim 12, in which, in relation to the employment of
      the software program as a result of which object types are identified which
      were not originally accounted for, modifying the software program in order to
25   take into account the object type so identified.

14.   A method as claimed in Claim 13, which includes modifying the base of object type information already gained.

15.   A method of performing a system reverse engineering process substantially as described in the specification with reference to and as illustrated in the accompanying diagrams.
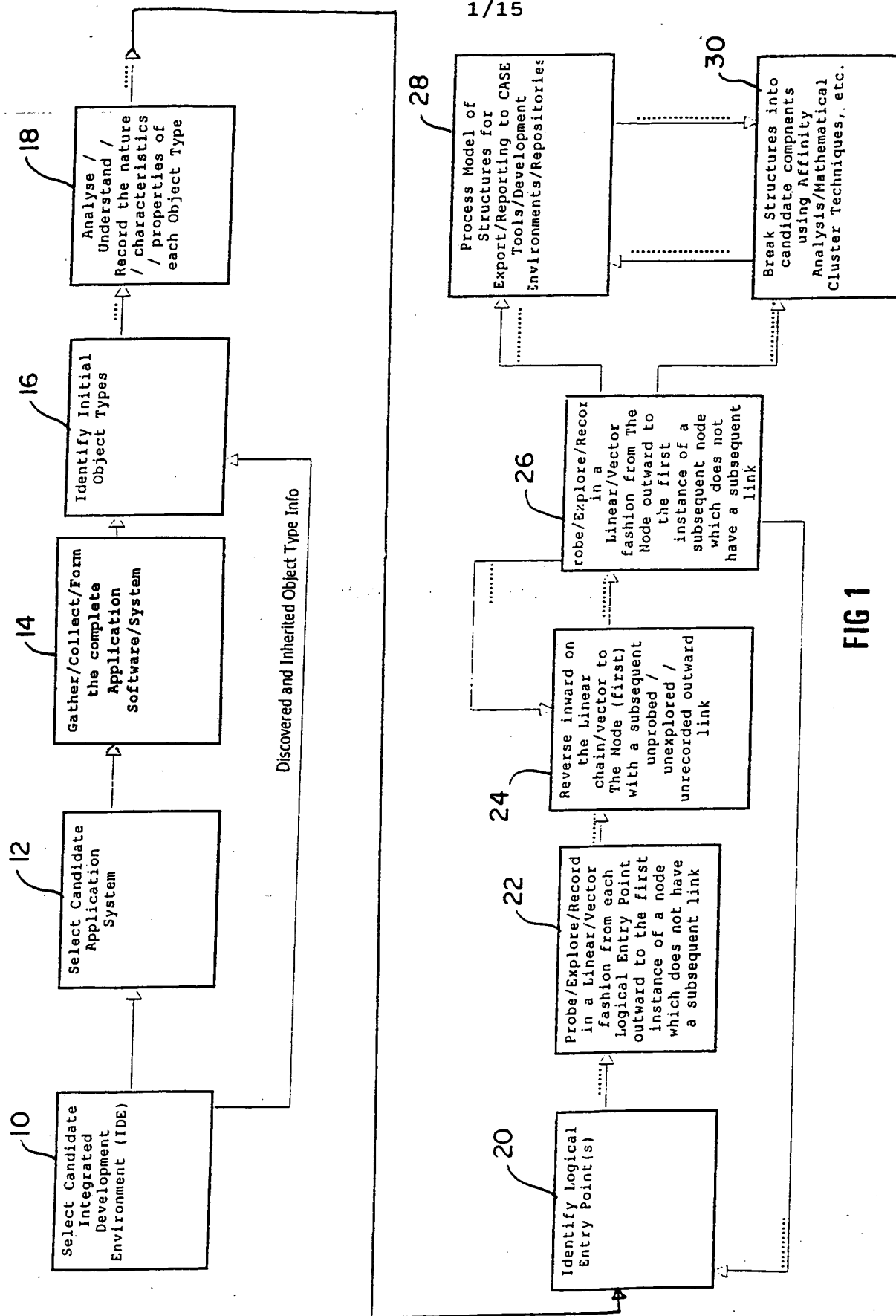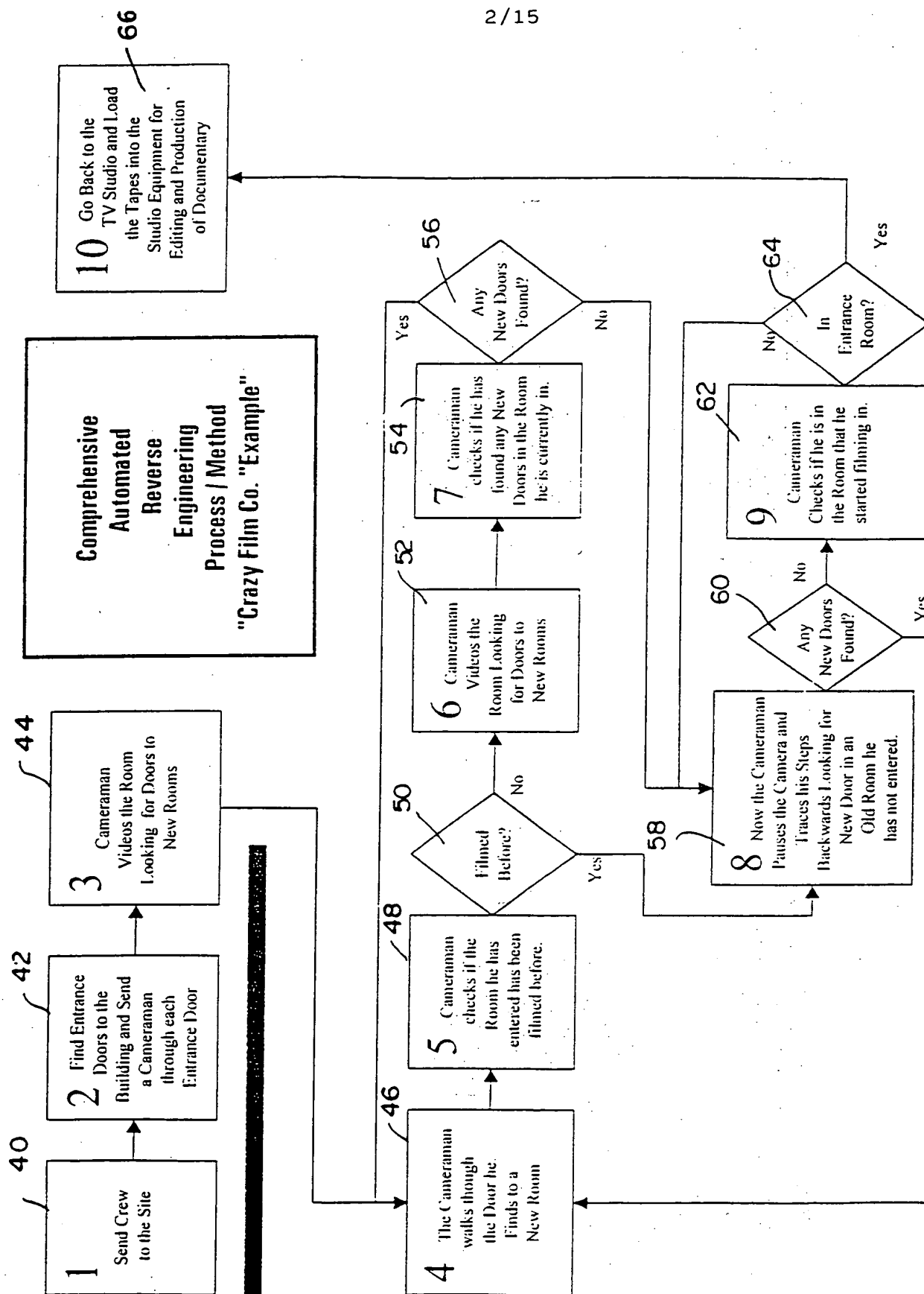
1/15

FIG 1

THIS PAGE BLANK (USPTO)

FIG 2

Comprehensive Automated
Reverse Engineering
Process / Method
Example

Limited to a single entry point of a
SQL Integrated Development
Environment.

SQL Data Definition
"Create Table" statement

FIG 3A

Overview

■ This example uses a non-systems
oriented example (Crazy Filming Co.)

■ The example steps are cross referenced
against an extract of a Comprehensive
Automated Reverse Engineering exercise
on a subset of a working Application
System.

FIG 3B

THIS PAGE BLANK (USPTO)

Identify an Application System

- An application system representing a complete business application must be located. **Step 1**

- This will be the source system for Comprehensive Automated Reverse Engineering. **Step 1**

FIG 3C

Gather all input files into VIRTUAL file (memory)

- For each physical input file **Step 2**
  - Read all lines from the input file into the VIRTUAL file **Step 2**

FIG 3D

THIS PAGE BLANK (USPTO)

Analyse, Parse and Record all Metadata in VIRTUAL file

- For each line in the VIRTUAL file, we iterate Step 3, 8 & 9
  - Assign the current line to storage variable
  - If line has "create table" then
    - Start looking for the "end of create table" marker by starting a new iteration, from the current position in the list. Step 3
    - Read the first of the new set of lines and reformat to ensure we have no unnecessary characters (tab stops and double spaces) in the line. Step 3

FIG 3E

Analyse, Parse and Record all Metadata in VIRTUAL file

- While the current line does not contain the "end of table" marker then: Step 4
  - Concatenate the current line with the storage variable Step 4
- Concatenate the current line, which contains the "end of table" marker, with the storage variable. Step 4
- If format of "create table" statement is correct then Step 4
  - Strip the table name from the table definition, and store each as a substring. Step 4

FIG 3F

THIS PAGE BLANK (USPTO)

09/868270

Analyse, Parse and Record all Metadata in VIRTUAL file

- Check if the table item exists in our nodal model, if it does not then: **Step 5**
  - Add the table name to the list of tables with a unique ID. **Step 6**
  - Trim known garbage from the ends of the table definition. **Step 7**
  - Find fields within the table defintion. **Step 7**

FIG 3G

Analyse, Parse and Record all Metadata in VIRTUAL file

- For each of the fields in the table definition line **Step 4 & 7**
    - Split the field into field name, field type, field storage format and field data entry constraint. **Step 4**
    - Check if it exists as nodal item linked to this table item, if it does not, then: **Step 5**
      > Add the field to the list of fields with a unique ID. (Record) **Step 6**
      > Add a tablefield link to the list of links for this field and current table. **Step 6**

FIG 3H

Analyse, Parse and Record all Metadata in VIRTUAL file

- Else if format of "create table" incorrect then
Step 4 & 7
  - Identify the incorrect statements line number, file name and location of file. **Step 4**
  - Check if this Unkown item exists as nodal item linked to the current file, if it does not, then: **Step 5**
    - Add the incorrect statement to the list of "Unkown" items with a unique ID **Step 6**
    - Add an unknownfile link to the list of links for this Unknown and respective file **Step 6**

FIG 3I

Analyse, Parse and Record all Metadata in VIRTUAL file

- End of has "create table" condition. **Step 8 & 9**
- If line has "create index" then
  - Check validity of create index statement. If it's invalid,
    - Identify the incorrect statements line number, file name and location of file. **Step 4**
    - Check if this Unkown item exists as nodal item linked to the current file, if it does not, then: **Step 5**
      - Add the incorrect statement to the list of "Unkown" items with a unique ID **Step 6**
      - Add a unknownfile link to the list of links for this Unknown and respective file **Step 6**

FIG 3J

THIS PAGE BLANK (USPTO)

Analyse, Parse and Record all Metadata in VIRTUAL file

- If the create index statement is valid:
    - Identify the field that the index is being created on. **Step 4**
    - Check if this "create index" item exists as nodal item linked to the identified field, if it does not exist: **Step 5**
        - Add the "create index" item to the nodal item list of indexes with a unique identifier. **Step 6**
        - If the field being indexed does not yet exist, add the new "create index" item to the list of items pending linkage. **Step 6**
        - Else Create an indexField link to the list of links that connects this "create index" to the affected field. **Step 6**

FIG 3K

Analyse, Parse and Record all Metadata in VIRTUAL file

- If the "create index" statement exists: **Step 5**
    - Add the incorrect or duplicate statement to the list of "Unkown" items with a unique ID **Step 6**
    - Add a unknownindex link to the list of links for this Unknown and its respective field. **Step 6**

■ End of VIRTUAL file iteration. **Step 10**

FIG 3L

Analyse, Parse and Record all Metadata in VIRTUAL file

- Iterate through list of items pending linkage. **Step 7**
  - If an the current unlinked item now has valid links:
    - create an item specific link item in the link list. **Step 6**
  - If the current unlinked item still has invalid link points:
    - Add an unknown item to the list of unknown items with a unique identifier. **Step 6**
    - Add a link from this unknown item to the file the unlinked item occurred in, and the line in that file. If there is a single side to the link, add an unknown item link from this unsuccessful link object to the known link point. **Step 6**
- End of items pending linkage iteration.

FIG 3M

Actual Data Manipulation Stages

FIG 3N

Original Statement In SQL Definition File

```
create table cheqmast
  (
      cheque no integer,
      cheque value decimal(9,2) not null,
      date chque date,
      practice char(7),
      delind char(1)
  );
create index i cheqmast on cheqmast (
    cheque no );
```

FIG 3O

After Concatenation and Whitespace Cleaning

```
create table cheqmast (cheque no
   integer,cheque value decimal (9,2) not
   null,date chque date,
practice char (7),delind char (1)
);
```

Validity is tested at this point.

FIG 3P

THIS PAGE BLANK (USPTO)

After Stage One Reformatting

(No space before comma)

```
create table cheqmast(cheque no
   integer,cheque value decimal(9,2) not
   null,date chque date,
practice char(7),delind char(1)
);
```

FIG 3Q

After Stage Two Reformatting

(Single space after open bracket)

```
create table cheqmast( cheque no
   integer,cheque value decimal( 9,2) not
   null,date chque date,
practice char( 7),delind char( 1)
);
```

FIG 3R

THIS PAGE BLANK (USPTO)

After Stage Three Reformatting

(Single space after open comma)
```
create table cheqmast( cheque no integer,
  cheque value decimal( 9, 2) not null,
  date chque date,
practice char( 7), delind char( 1)
);
```

FIG 3S

After Stage Four Reformatting

(Single space before closing bracket)
```
create table cheqmast( cheque no integer,
  cheque value decimal( 9, 2 ) not null,
  date chque date,
practice char( 7 ), delind char( 1 )
);
```

FIG 3T

THIS PAGE BLANK (USPTO)

Split Table Name and Definition

```
create table cheqmast( cheque no integer,
   cheque value
decimal( 9, 2 ) not null, date chque date,
practice char( 7 ), delind char( 1 )
);
```
And leaves us with:

Table name   : cheqmast
Table definition   : (cheque_no integer, cheque_value
                       decimal(9, 2) not null, date_chque date,
                       practice char(7), delind char(1)
                       );

The table node is stored.

FIG 3U

After Trim known garbage from the ends of the definition

```
cheque no integer, cheque value decimal( 9, 2
   ) not null, date chque date,
practice char( 7 ), delind char( 1 )
```

FIG 3V

THIS PAGE BLANK (USPTO)

After Splitting Fields

```
cheque no integer
cheque value decimal( 9, 2 ) not null
date chque date
practice char( 7 )
delind char( 1 )
```

FIG 3W

Find data types, formats and constraints, finds the following

```
cheque value decimal( 9, 2 ) not null
^----------^ ^------^  ^--^   ^-------^
   Name        Type    Format  Constraint
```

Which is stored as:

Field Name	: cheque_value
Field Type	: decimal
Field Storage Format	: 9, 2
Field Data Entry Constraint	: not null

Each of the field nodes are stored and linked, using a link node, to their respective table node and the next table node is found and analysed.

FIG 3X

THIS PAGE BLANK (USPTO)

Analyse "create index" statement

```
create index i_cheqmast on cheqmast(cheque_no);
^----------^^---------^   ^-----^ ^------^
  Statement    Name         Table    Field
```

Which is stored as:

Index Name   : i_cheqmast
Index Table   : cheqmast
Field to index on   : cheque_no

If the table cheqmast does not yet exist, or does not yet have a field by the name cheque_no, the "create index" object is created and placed in the "to link" list.

If the table and field both exist, the "create index" object is created and an index-field link is added to the list of links.

FIG 3Y

THIS PAGE BLANK (USPTO)

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference F14839	**FOR FURTHER ACTION**	see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.
International application No. PCT/IB 99/01989	International filing date *(day/month/year)* 14/12/1999	(Earliest) Priority Date *(day/month/year)* 18/12/1998
Applicant GORDON, Graham, Paul et al.		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of _____3_____ sheets.

[X] It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

   a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

      [ ] the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

   b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

      [ ] contained in the international application in written form.

      [ ] filed together with the international application in computer readable form.

      [ ] furnished subsequently to this Authority in written form.

      [ ] furnished subsequently to this Authority in computer readble form.

      [ ] the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

      [ ] the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. [ ] **Certain claims were found unsearchable** (See Box I).

3. [ ] **Unity of invention is lacking** (see Box II).

4. With regard to the **title**,

   [X] the text is approved as submitted by the applicant.

   [ ] the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

   [X] the text is approved as submitted by the applicant.

   [ ] the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.   1_____

   [X] as suggested by the applicant.   [ ] None of the figures.

   [ ] because the applicant failed to suggest a figure.

   [ ] because this figure better characterizes the invention.

THIS PAGE BLANK (USPTO)

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CANFORA G ET AL: "AN IMPROVED ALGORITHM FOR IDENTIFYING OBJECTS IN CODE" SOFTWARE PRACTICE & EXPERIENCE,GB,JOHN WILEY & SONS LTD. CHICHESTER, vol. 26, no. 1, 1 January 1996 (1996-01-01), pages 25-48, XP000655430 ISSN: 0038-0644 page 25, line 1 -page 34, line 11 --- -/--	1,6,10

[X] Further documents are listed in the continuation of box C.   [ ] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 April 2000	03/05/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Kingma, Y

Form PCT/ISA/210 (second sheet) (July 1992)

THIS PAGE BLANK (USPTO)

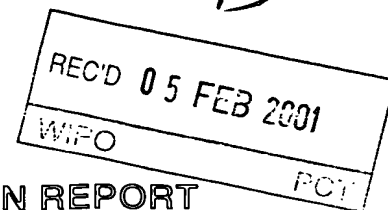**C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	GALL H ET AL: "BALANCING IN REVERSE ENGINEERING AND IN OBJECT-ORIENTED SYSTEMS ENGINEERING TO IMPROVE REUSABILITY AND MAINTAINABILITY" , PROCEEDINGS OF THE ANNUAL INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC),US,LOS ALAMITOS, IEEE COMP. SOC. PRESS, VOL. CONF. 18, PAGE(S) 35-42 XP000611144ISBN: 0-8186-6707-9 page 35, left-hand column, line 1 -page 37, left-hand column, line 20	1,4,8

THIS PAGE BLANK (USPTO)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference F14839	FOR FURTHER ACTION	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/IB99/01989	International filing date *(day/month/year)* 14/12/1999	Priority date *(day/month/year)* 18/12/1998

International Patent Classification (IPC) or national classification and IPC

G06F9/44

Applicant

GORDON, Graham, Paul et al.

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 6 sheets, including this cover sheet.

    ☒ This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

    These annexes consist of a total of 5 sheets.

3. This report contains indications relating to the following items:

    I   ☒ Basis of the report
    II   ☐ Priority
    III ☒ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
    IV   ☐ Lack of unity of invention
    V   ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations suporting such statement
    VI   ☐ Certain documents cited
    VII ☐ Certain defects in the international application
    VIII ☒ Certain observations on the international application

Date of submission of the demand 03/07/2000	Date of completion of this report 31.01.2001
Name and mailing address of the international preliminary examining authority: European Patent Office D-80298 Munich Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Authorized officer Jaedicke, M Telephone No. +49 89 2399 2357

I.   Basis of the report

1. This report has been drawn on the basis of *(substitute sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments (Rules 70.16 and 70.17).)*:
    Description, pages:

1,3-12	as originally filed				
2,2a	as received on	04/12/2000	with letter of	01/12/2000

Claims, No.:

1-6	as received on	04/12/2000	with letter of	01/12/2000

Drawings, sheets:

1/15-15/15	as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

    These elements were available or furnished to this Authority in the following language: , which is:

    ☐   the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).

    ☐   the language of publication of the international application (under Rule 48.3(b)).

    ☐   the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

    ☐   contained in the international application in written form.

    ☐   filed together with the international application in computer readable form.

    ☐   furnished subsequently to this Authority in written form.

    ☐   furnished subsequently to this Authority in computer readable form.

    ☐   The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

    ☐   The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. The amendments have resulted in the cancellation of:

THIS PAGE BLANK (USPTO)

☐ the description,   pages:

☐ the claims,   Nos.:

☐ the drawings,   sheets:

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)):

(Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)

6. Additional observations, if necessary:

III. Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

1. The questions whether the claimed invention appears to be novel, to involve an inventive step (to be non-obvious), or to be industrially applicable have not been examined in respect of:

☐ the entire international application.

☒ claims Nos. 6.

because:

☐ the said international application, or the said claims Nos. relate to the following subject matter which does not require an international preliminary examination (*specify*):

☒ the description, claims or drawings (*indicate particular elements below*) or said claims Nos. 6 are so unclear that no meaningful opinion could be formed (*specify*):
see separate sheet

☐ the claims, or said claims Nos. are so inadequately supported by the description that no meaningful opinion could be formed.

☐ no international search report has been established for the said claims Nos. .

2. A meaningful international preliminary examination report cannot be carried out due to the failure of the nucleotide and/or amino acid sequence listing to comply with the standard provided for in Annex C of the Administrative Instructions:

☐ the written form has not been furnished or does not comply with the standard.
☐ the computer readable form has not been furnished or does not comply with the standard.

V. Reasoned statem nt under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

THIS PAGE BLANK (USPTO)

1. Statement

Novelty (N)	Yes:	Claims	1-5
	No:	Claims	
Inventive step (IS)	Yes:	Claims	1-5
	No:	Claims	
Industrial applicability (IA)	Yes:	Claims	1-5
	No:	Claims	

2. Citations and explanations
   see separate sheet

VIII. Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made:
see separate sheet

THIS PAGE BLANK (USPTO)

R It m III
Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

1. Claim 6 is so unclear that no meaningful opinion can be established - see section VIII, point 3.

Re Item V
Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Reference is made to the following document:

    D1: GALL H ET AL: 'BALANCING IN REVERSE ENGINEERING AND IN OBJECT-ORIENTED SYSTEMS ENGINEERING TO IMPROVE REUSABILITY AND MAINTAINABILITY' , PROCEEDINGS OF THE ANNUAL INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC),US, LOS ALAMITOS, IEEE COMP. SOC. PRESS, VOL. CONF. 18, PAGE(S) 35-42, XP000611144, ISBN: 0-8186-6707-9.

2. In the following, claim 1 is interpreted as explained in section VIII, point 2.

    The prior art document D1 discloses in chapter 2 a reverse engineering process that is used to generate design documents from an existing software system.

    One of the differences between the method of performing a system reverse engineering process on an application system according to claim 1 and the closest prior art document D1 is that the claimed method uses the aid of a 'suitable programmed processing system which is provided with a data base of object types (...) which fall in groups that include process or activity control elements, data management elements and interface elements'. While the prior art discloses that some steps of the reverse engineering can be performed automatically, it does not teach or fairly suggest the use of such a data base of object types.

Hence, the method of claim 1 meets the requirements of Article 33 PCT in respect of novelty, inventive step and industrial applicability.

Re Item VIII

Certain observations on the international application

1. The plurality of independent method claims 1 and 6 makes it unclear what the applicant considers to be the features which are necessary to define the invention for which protection is sought. Hence, the independent claims as a whole are unclear (Article 6 PCT). In particular, in the present case it is not necessary to have more than one independent claim in any one category.

2. Claims 1 and 3 are unclear (Article 6 PCT), because the meaning of the expression 'nature' (of object types, or nodes and links) to which these claims refer is not clear. For the purposes of section V, this term is simply interpreted as being redundant to the expression 'characteristics and properties', i.e. it is ignored.

3. Claim 6 is unclear (Article 6 PCT), because it specifies a method of performing a system reverse engineering process merely by references to the description and the drawings. According to Rule 6.2(a) PCT, claims should not contain such references except where absolutely necessary, which is not the case here.

   Moreover, claim 6 specifies that the method is "substantially" as described in the description. This vague expression renders the definition of the subject matter of the claimed method unclear.

THIS PAGE BLANK (USPTO)

structures are further complicated insofar as two or more links can extend from a
node and by keeping in mind that a network structure could include millions of nodes
and links, it will be appreciated that very intricate structures can result. The
individual nodes and links referred to essentially are object instances and messages

5   disposed between the object instances, representing activity and data elements which
are associated with the operation of the system, for carrying out its required purpose
or purposes.

In order to apply a reverse engineering process to an application system, it is required
to obtain a full understanding of the system, i.e. an understanding of the operation

10   of the network structure forming the system, in order to permit the system to be
reverse engineered into a format which permits a model of the system to be created
with the aid of a suitable CASE tool (Computer Aided Software/Systems Engineering)
or any other categories of suitable visualisation tools. The examination of an
application system in order to acquire an understanding of the system conventionally

15   involves an overall consideration of the system and then progressively delving into
the system from a number of predetermined starting points, delving deeper and
deeper into the system until the required understanding is acquired. This generally
requires the cooperation of a team of suitably qualified systems engineers who will
cooperate with one another and add their knowledge together until the required level

20   of understanding of the system is acquired, which then permits reverse engineering.
This examination system is well known and produced desired results in relation to
relatively simpler and less dynamically changing application systems where systems
engineers could acquire a sufficient overall picture of the system to permit reverse
engineering thereof, but in relation to more complex and dynamic systems this

25   method of examination became too complex and time consuming and, as such,
impractical.

The more complex application systems requiring reverse engineering therefore cannot
be economically salvaged, even with the aid of software programs assisting with the
examination process as above envisaged. As such, it is an object of this invention

30   to provide an improved method of performing a system reverse engineering process

THIS PAGE BLANK (USPTO)

CLAIMS

1.   A method of performing a system reverse engineering process, which includes
      the steps of :

      identifying the application system that requires reverse engineering and
5   gathering the entire system and identifying the development environment
      associated with the system;

      identifying initial object types that can serve as starting points from where an
      examination of the system can be initiated and analysing the nature,
      characteristics and properties of each object type;

10   identifying entry points, in the form of object instances of certain object types,
      for entering the system to carry out the examination of the system;

      examining from selected entry points the network structure forming the
      application system by tracking chains of nodes and links, each chain being
      tracked until the instance of a node that does not have a link or the return of
15   the chain to a previously examined node, then reverse tracking the chain to a
      node from which another chain extends and selectively tracking said other
      chain and continuing the process until all the chains within the network
      structure have been tracked, the tracking of the chains including an
      examination of each node and link and a recordal of information so gathered;
20   and

      from the information gathered by the network examination, formatting the
      information gathered into a form in which it represents the application system
      in a usable form.

2.   A method as claimed in Claim 1, in which the development environment
25   identified is an integrated development environment.

THIS PAGE BLANK (USPTO)

3.   A method as claimed in Claim 1 or Claim 2, which includes in identifying the development environment, identifying aspects of the development environment selected from a group including program language and syntax used, the mechanisms of storage of data, the interface of the above, component libraries and code management systems.

4.   A method as claimed in any one of Claims 1 to 3, in which the object types identified fall into multiple groups, respective groups including process or activity control elements, data management elements and interface elements.

5.   A method as claimed in any one of the preceding claims, which includes examining the network structure by an examination of each node and each link in the structure to the extent that the nature, characteristics and properties of each node and each link can be associated with object types through analysis and understanding thereof and gathering and recording all the information of each node and each link.

6.   A method as claimed in any one of the preceding claims, which includes identifying entry points that comprise the nodes from which a complete examination of the entire network structure of the application system can be initiated.

7.   A method as claimed in any one of the preceding claims, which includes examining the network structure by the complete tracking of each chain to its termination or return to an earlier node in the chain, before a further chain is selected and tracked.

8.   A method as claimed in any one of the preceding claims, in which the tracking of chains within the examination of the network structure includes a comparison and classification of nodes and links as object instances to establish whether they confirm with known or unknown object types.

THIS PAGE BLANK (USPTO)

9. A method as claimed in any one of the preceding claims, which includes formatting the information gathered into a format in which the information can be exported/reported to at least one of a predetermined computer aided software/systems engineering tool, a development environment and a repository, which will enable the creation of a model of the application system.

10. A method as claimed in Claim 9, in which formatting includes breaking structures into candidate components by using affinity analysis and mathematical clustering techniques.

11. A method as claimed in any one of the preceding claims, which includes employing any one of software and hardware for assisting with the identification of object instances of object types and the analysis of the nature, characteristics, attributes, operations and properties of each object instance and object type identified, the identification of entry points for entering the system to carry out the examination of the network structure forming the system, the actual examination of the network structure and the formatting of information gathered by the examination of the network structure into a usable form.

12. A method as claimed in Claim 11, which includes creating a software program or entire software system which can be employed in respect of the particular application system and associated development environment being reverse engineered.

13. A method as claimed in Claim 12, in which, in relation to the employment of the software program as a result of which object types are identified which were not originally accounted for, modifying the software program in order to take into account the object type so identified.

14.   A method as claimed in Claim 13, which includes modifying the base of object
        type information already gained.

15.   A method of performing a system reverse engineering process substantially as
        described in the specification with reference to and as illustrated in the
5   accompanying diagrams.

THIS PAGE BLANK (USPTO)

structures are further complicated insofar as two or more links can extend from a node and by keeping in mind that a network structure could include millions of nodes and links, it will be appreciated that very intricate structures can result. The individual nodes and links referred to essentially are object instances and messages disposed between the object instances, representing activity and data elements which are associated with the operation of the system, for carrying out its required purpose or purposes.

In order to apply a reverse engineering process to an application system, it is required to obtain a full understanding of the system, i.e. an understanding of the operation of the network structure forming the system, in order to permit the system to be reverse engineered into a format which permits a model of the system to be created with the aid of a suitable CASE tool (Computer Aided Software/Systems Engineering) or any other categories of suitable visualisation tools. The examination of an application system in order to acquire an understanding of the system conventionally involves an overall consideration of the system and then progressively delving into the system from a number of predetermined starting points, delving deeper and deeper into the system until the required understanding is acquired. This generally requires the cooperation of a team of suitably qualified systems engineers who will cooperate with one another and add their knowledge together until the required level of understanding of the system is acquired, which then permits reverse engineering. This examination system is well known and produced desired results in relation to relatively simpler and less dynamically changing application systems where systems engineers could acquire a sufficient overall picture of the system to permit reverse engineering thereof, but in relation to more complex and dynamic systems this method of examination became too complex and time consuming and, as such, impractical.

The known approach to system reverse engineering was identified by GALL H et al, under the title "Balancing in Reverse Engineering and in Object - Oriented Systems Engineering to Improve Reusability and Maintainability", during proceedings of the

AMENDED SHEET

THIS PAGE BLANK (USPTO)

Annual International Computer Software and Applications Conference (COMPSAC), US, Los Alamitos, IEEE Comp. Soc. Press, Vol. Conf. 18, Page(s) 35-42, XP 000611144, ISBN: 0-8186-6707-9 and particularly also in a reference cited therein and identified as P. Benedusi, A. Cimitile, and U. de Carlini. Reverse Engineering Processes, Design Document Production, and Structure Charts. The Journal of Systems and Software, 19 (3) : 225-245, November 1992.

The above references clearly identify known system reverse engineering processes as goal orientated processes. A goal orientated process involves only an examination of predetermined parts of a system to be reverse engineered, particularly to extract and understand these parts to the extent that they can be reverse engineered into a format which accommodates a predetermined goal. Such processes clearly are limited processes which accommodate only specific goals. This limitation of reverse engineering to merely achieve goals has always been considered essential in order to render reverse engineering practically possible and particularly when more complex systems are involved, even merely to reverse engineer to the extent of achieving defined goals has proven to be extremely difficult

The more complex application systems requiring reverse engineering therefore cannot be economically salvaged, even with the aid of software programs assisting with the examination process as above envisaged. As such, it is an object of this invention to provide an improved method of performing a system reverse engineering process

THIS PAGE BLANK (USPTO)

CLAIMS

1. A method of performing a system reverse engineering process on an application system, which includes the steps of:

   gathering the entire application system that requires reverse engineering and identifying each development environment associated with the application system; and

   with the aid of a suitably programmed processing system, which is provided with a data base of object types in respect of which the nature, characteristics and properties are known and which fall in groups that include process or activity control elements, data management elements and interface elements, performing the steps of:

   identifying the object types, as determined by each development environment identified as being associated with the application system, that can serve as starting points from where an examination of the application system can be initiated;

   identifying entry points, in the form of object instances of object types identified to serve as starting points from where an examination of the application system can be initiated, for entering the system to carry out an examination of the system;

   examining from selected entry points the network structure forming the application system by tracking chains of nodes and links, each chain being tracked until the instance of a node that does not have a link or the return of the chain to a previously examined node, then reverse tracking the chain to a node from which another chain extends and selectively tracking said other chain and

continuing the process until all the chains within the network structure have been tracked completely, the tracking of the chains including an examination of each node and each link in the network structure, to the extent that the nature, characteristics and properties of each node and each link can be associated with
5   object types through analysis and understanding thereof, and gathering and recording all the information of each node and each link; and

from the information gathered and recorded by the examination of the network structure forming the application system, formatting the information gathered into a form in which it represents the application system in a usable form.

10   2.   A method as claimed in Claim 1, which includes, in identifying each development environment associated with the application system, identifying aspects of each development environment selected from a group including programming language and syntax used, the mechanisms of storage of data, the interface of the above, component libraries and code management systems.

15   3.   A method as claimed in Claim 1 or Claim 2, in which the examination of each node and each link in the chain of the network structure forming the application system, while tracking the chains, includes a comparison and classification of nodes and links as object instances of object types to establish whether they conform with known object types included in the database of the processing
20   system used, or unknown object types and where they conform with unknown object types, identifying the nature, characteristics and properties of these unknown object types and then including them in the said database to become known object types.

4.   A method as claimed in any one of the preceding claims, which includes
25   formatting the information gathered and recorded, by the examination of the network structure forming the application system, into a format into which the

AMENDED SHEET

information can be exported/reported to at least one of a computer aided software/systems engineering tool, a development environment and a repository, which will enable the creation of a model of the application system.

5. A method as claimed in Claim 4, in which formatting includes breaking structures into candidate components by using affinity analysis and mathematical clustering techniques.

6. A method of performing a system reverse engineering process substantially as described in the specification with reference to and as illustrated in the accompanying diagrams.